

In the claims

1. (Currently Amended) A method of creating extensible and repurposable content, comprising the steps of:

- (a) creating a document type definition (DTD);
- (b) generating a form-based data entry interface based, at least in part, on the DTD, wherein the form-based data entry interface is implemented as a portable document format accessible via a portable document format compatible application program that is incapable of saving data in the form of extensible markup language (XML);
- (c) inputting data into the form-based data entry interface that is accessible via the application program;
- (d) applying an extensible stylesheet language (XSL) construct to the data that is input at step (c) by accessing the form-based data entry interface via a script based plug-in to the application program.; and
- (e) through application of the XSL, saving the data in the form of XML consistent with the DTD to produce a first XML file while contemporaneously saving the data in the form in the portable document format to produce a first portable document format file;
- (f) after saving the data and closing the form-based data entry interface, accessing the form-based data entry interface and data in the portable document format via the portable document format compatible application program;
- (g) editing the data through the form-based data entry interface via the application program;
- (h) re-applying the XSL construct to the edited data by again accessing the form-based data entry interface via the script based plug-in to the application program; and
- (i) through application of the XSL, saving the edited data in the form of XML consistent with the DTD to replace the first XML file while contemporaneously saving the edited data in the form in the portable document format to replace the first portable document format file.

2. (Original) The method of claim 1, wherein the DTD defines an aspect of a business.

3. (Original) The method of claim 1, wherein the content is represented by the DTD.

4. (Cancelled)

5. (Original) The method of claim 1, the data saved at step (e) is saved in world wide web consortium (W3C) compliant XML.

6. (Original) The method of claim 1, further comprising employing the data saved at step (e) for different purposes.

7. (Original) The method of claim 5, wherein the data saved at step (e) is employed to generate a graphics product.

8. (Original) The method of claim 5, wherein the data saved at step (e) is employed to generate a text-to-voice information.

9. (Original) The method of claim 5, wherein the data saved at step (e) is employed as content for a wireless message.

10. (Currently Amended) A method of preparing an entry in a directory having a plurality of separately prepared entries, the method comprising the steps of:

(a) accessing an extensible markup language (XML) file in a database that is at least partially representative of the entry to be included in the directory, the XML file being based on a document type definition (DTD) consistent with the directory;

(b) launching a first software program, the first software program presenting a form that is displayed on a computer, wherein the first software program has a native file saving format that is different from a format of the XML file and wherein the first software program is incapable of opening and saving data that is in the format of the XML file;

(c) populating selected fields in the form by transforming the XML file using (i) an extensible stylesheet transformation (XSLT) file and (ii) an application programming interface (API) exposed by the first software program where a script-based plug-in provides information of the XML file that is transformed by the XSLT to the first software program for inclusion in a first portable document format file;

(d) at least one of modifying the selected fields and adding data to fields other than the selected fields;

(e) spawning a graphics page using the first software program and displaying the entry; and

(f) saving the entry as an XML file by using the XSLT file and the API wherein the script-based plug-in provides information of the first portable document format file from the first software program to the XSLT, and contemporaneously saving the first portable document format file;

(g) after closing the first portable document format file, then accessing the first portable document format file to edit the graphics page within the first software program;

(h) re-applying the XSLT to the edited graphics page by again accessing the information via the script based plug-in to the first software program; and

(i) through application of the XSL, saving the edited graphics page to replace the XML file while contemporaneously saving the edited graphics page in the portable document format to replace the first portable document format file.

11. (Original) The method of claim 10, further comprising editing components of the entry using a second software program.

12. (Original) The method of claim 10, further comprising saving the entry as an encapsulated postscript (EPS) file.

13. (Original) The method of claim 10, wherein the directory comprises a dining guide.

14. (Original) The method of claim 13, wherein the entry comprises an advertisement in the dining guide.

15. (Original) The method of claim 10, further comprising selecting an entry style.

16. (Original) The method of claim 15, wherein the entry style is described by a universal directory advertising code (UDAC).

17. (Original) The method of claim 10, further comprising accessing the XML file via a copy ID.

18. (Currently Amended) A method of publishing, comprising the steps of:

(a) saving first information associated with a graphics design in a first database as a first portable document format file;

(b) assigning a copy ID to the first portable document format file including the first information;

(c) transferring the first portable document format file including the first information from the first database to a second database;

(d) accessing the first portable document format file including the first information via the second database and via the copy ID and populating selected fields of a form that is implemented as a portable document format and displayed on a computer implementing a first portable document format compatible computer application that is ~~wherein the form~~ is incapable of saving the information as an extensible markup language (XML) file;

(e) at least one of (i) modifying the first information and (ii) adding second information to fields other than the selected fields via the form being implemented as a portable document format;

(f) spawning a graphics page on the computer that incorporates the first and second information in a pre-selected style for the graphics design, wherein the graphics

page is further implemented via the first portable document format compatible computer application that is incapable of saving the information as an XML file;

(g) editing graphics components of the graphics design resulting in an edited graphics design; ~~and~~

(h) saving the edited graphics design as an XML file by applying an extensible stylesheet language transformation (XSLT) file to the first and second information by accessing the first portable document format compatible computer application via a script-based plug-in;

(i) after saving the edited graphics design and closing the form, accessing the form and edited graphics design in the portable document format via the portable document format compatible application program;

(j) editing the edited graphics design through the form-based data entry interface via the application program;

(k) re-applying the XSL construct to the edited graphics design by again accessing the application program via the script based plug-in; and

(l) through application of the XSL, saving the edited graphics design in the form of XML to replace the XML file while contemporaneously saving the edited graphics design in the form in the portable document format to replace the first portable document format file.

19. (Original) The method of claim 18, wherein the graphics design is an advertisement.

20. (Original) The method of claim 18, wherein the first information comprises at least one of a name of a restaurant, a restaurant address, and an advertisement style.

21. (Original) The method of claim 20, wherein the advertisement style is described as a universal directory advertising code (UDAC).

22. (Original) The method of claim 18, wherein the first database is accessible primarily by salespeople.

23. (Original) The method of claim 18, wherein the second database is accessible primarily by a graphics artist.

24. (Cancelled)

25. (Currently Amended) The method of claim 18, wherein the step of saving comprises transforming a first file format that the first and second information is saved within into XML using a predetermined XSLT file.

26. (Original) The method of claim 25, wherein the first file format is XFDF.

27. (Original) The method of claim 18, further comprising saving the edited graphics design as an encapsulated postscript (EPS) file.

28. (Original) The method of claim 27, further comprising forwarding the EPS file to a printer.

29. (Currently Amended) A system for generating an entry for a directory that comprises a plurality of entries, the system comprising:

- (a) a first workstation operable to receive first information related to the entry;
- (b) a first database for storing the first information;
- (c) a second database that periodically receives the first information from the first database; and

- (d) a second workstation, the second workstation being operable to access the second database,

wherein the second workstation is operable to (i) launch a first software program, the first software program presenting a form that is displayed on the second workstation, wherein the first software program has a native file saving format that is different from a format of an XML file and wherein the first software program is incapable of saving a file in the format on an XML file, (ii) at least one of modify selected fields in the form

and add data to fields other than the selected fields, (iii) spawn a graphics page using the first software program to display the entry, and (iv) save the entry as an XML file by applying an extensible stylesheet language transformation (XSLT) file to the data of the entry, wherein the XSLT is applied by accessing the first software program via a script-based plug-in, wherein the first software program saves the graphics page as a portable document format file contemporaneously with the XSLT creating an XML file, wherein the saved graphics page is re-opened by the first software program accessing the portable document format file to edit the graphics page, wherein the XSLT is re-applied to the edited graphics page to replace the XML file with a new XML file while the first software program contemporaneously saves the edited graphics page to replace the portable document format file with a new portable document format file.

30. (Cancelled).

31. (Cancelled)

32. (Original) The system of claim 29, wherein the second workstation is further operable to save the entry as an encapsulated postscript (EPS) file.

33. (Original) The system of claim 29, wherein the directory is a dining guide.

34. (Original) The system of claim 33, wherein the entry comprises an advertisement in the dining guide.

35. (Original) The system of claim 29, wherein the first database is accessible primarily by a salesperson.

36. (Original) The system of claim 29, wherein the second database is accessible primarily by a graphics artist.